Efficient quarantining of scanning worms: optimal detection and coordination

A. Ganesh, D. Gunawardena, P. Key, L. Massoulié
Microsoft Research, Cambridge, U.K.
Email: {ajg,dinang,peter.key,lmassoul}@microsoft.com

J. Scott
EE & CS Department, UC Berkeley
jhs@ocf.berkeley.edu

***Abstract*— Current generation worms have caused considerable damage, despite their use of unsophisticated scanning strategies for detecting vulnerable hosts. A number of adaptive techniques have been proposed for quarantining hosts whose behaviour is deemed suspicious. Such techniques have been proven to be effective against fast scanning worms. However, worms could evade detection by being less aggressive. In this paper we consider the interplay between worm strategies and detection techniques, which can be described in game-theoretic terms.**

We use epidemiological modelling to characterise the outcome of the game (the pay-off function), as a function of the strategies of the worm and the detector. We design detection rules that are optimal against scanning worms with known characteristics. We then identify specific detection rules that are close to optimal, in some mathematically precise sense, against *any* scanning worm. Finally, we design methods for coordinating information among a set of end-hosts, using Bayesian decision theory. We evaluate the proposed rules using simulations driven by traces from a corporate environment of 600 hosts, and assess the benefits of coordination.

I. Introduction

A worm is a self-propagating malicious program that exploits security vulnerabilities and does not require user action to propagate. Weaver et al [16] give a general taxonomy of worms. They list target discovery – the mechanism by which a worm discovers vulnerable hosts – as one of the critical factors of a worm's strategy. Current generation worms have typically used simple strategies, such as random or sequential scanning of the address space, to discover vulnerable hosts. With the address space the IPv4 namespace, the Code Red worm [11] managed to infect some 360,000 hosts in 24 hours, Blaster infected around 150,000 hosts in days while Slammer [10] infected 75,000 hosts in less than 15 minutes.

Several techniques to detect and contain worms have been proposed and implemented. Many of these combine some form of anomaly detection with either rate-limiting or quarantining the host. A simple example is Williamson's throttle [17], [15], which combines a rate throttle with an LRU whitelist cache to limit the rate of connections to new hosts to 1Hz. Such rate-limit throttles can cause problems for high-load machines such as servers, or for peer-to-peer applications. Schechter et al. [13] use a credit throttle combined with a reverse sequential hypothesis test that looks at both failures and successes. A similar idea was proposed to deal with port-scanning in general [6]. Anomaly detection schemes have the advantage that they are exploit-neutral, and so they work for zero-day worms (worms targeting an unknown vulnerability). They can be implemented by subnet-level intrusion detectors [13] or as part of network level intrusion detectors at gateways, or on the end-hosts themselves. Honey-pots and populating the address space with dark addresses are other ways to detect malicious scanning attempts.

Detecting anomalous behavior is just one line of defence. Another technique is packet payload monitoring, used for example in Autograph [7]; by detecting a worm signature, worm packets can be filtered out at network routers. This relies on detection, identification and probably infection, so has some difficulties with zero-day exploits, as well as with polymorphic worms. A promising alternative approach based on using a collaborative system for broadcasting self-certifying alerts [4] has also been proposed recently. Ideally security vulnerabilities are themselves prevented, an area the programming language community is addressing. But while vulnerabilities do exist, we would like to detect worms and stop their spread. Despite the different approaches described above, we believe that detecting anomalous scanning behaviour continues to be a useful weapon against worms, and that in practice multi-faceted defence has advantages.

More sophisticated scanning strategies involve harvesting known addresses, e.g., by using address information on the local host (such as IP or layer 2 information available through local host files, netstat statistics, arp tables etc, or addresses in local address books). Such 'topological' worms are extremely difficult to detect since malicious behavior may be hard to distinguish from normal behavior, and are outside the scope of this paper.

Our focus in this paper is on detecting anomalies due to worm scanning behaviour, based on end-system monitoring and co-ordinated responses. In section II we give an analytic treatment of countermeasures based on epidemic models and use this to analyse particular throttling schemes. In section III we look at optimal detection mechanisms, based on the CUSUM technique which is well-known in the control theory literature. These detection mechanisms have close links to the reverse sequential likelihood ratio test of [6], [13], though we specifically focus on failures to avoid a worm gaming the mechanism. We view the interaction between the worm and detector as a game; the detector chooses a strategy for quarantining, the worm chooses a scanning rate and the pay-off is the speed of spread (the growth exponent of the epidemic). This enables us to determine the Minimax solution, i.e. the best detector against the worst worm. We further derive suboptimal practical policies. In section V, we describe how to coordinate information amongst a set of hosts by using a Bayesian decision theoretic framework, in order to improve the detection of misbehaving hosts. Note that this is different from an aggregate detector: we are pooling individual information and then feeding that information back to identify the affected hosts, something an aggregate detector cannot do unless it sees all the information available to individual hosts. We use trace-driven simulations to assess the performance of worms and countermeasures, with and without coordination. For our data we find that without coordination the optimal (slow-scanning) worm can still infect a significant fraction of the network, whereas co-ordination can further limit the spread.

II. Epidemiological modelling of countermeasures

The aim of this section is to describe the outcome of a worm infection in a network with countermeasures deployed at every host. We focus on two types of countermeasures: (a) throttling schemes, whereby the rate at which a host can make new connections is reduced, and (b) quarantining schemes, whereby a host, when deemed infected, is denied further access to other hosts. Conveniently, the effect of both types of countermeasures can be envisaged in a single framework, which we now describe.

A. General framework

Consider an address space of size Ω populated by N hosts sharing some vulnerability which is exploited by the worm. When host j becomes infected, it scans the address space at rate v_j and infects any vulnerable host it locates. We use the following notation for key parameters throughout this section: $\alpha = N/\Omega$ is the fraction of the scanned address space which is populated by addresses corresponding to vulnerable hosts; v is the scanning rate of infected hosts; $r(t)$ is the fraction of vulnerable hosts that has been infected by time t. Formally, we consider a sequence of such systems indexed by N, and assume that α and v remain constant. Suppose first that no countermeasures are in place. If we assume that the scan times are described by a Poisson process, then $r(t)$ evolves as a Markov process. As N tends to infinity, the sequence of Markov process describing the fraction of infected hosts converges to the solution of a differential equation. This follows by a straightforward application of Kurtz's theorem [14, Theorem 5.3]. For background on differential equation models of epidemics, see [5]; for applications to Internet worms, see [18] and references therein. We shall use differential equations to model the evolution of r over time, including the impact of countermeasures. While we omit proofs, it should be understood that these models arise in the large population limit described formally above, and that their use can be rigorously justified using Kurtz's theorem.

In addition to the above parameters, we shall summarize the effect of the countermeasure by a reduction factor, $\theta(t)$, which affects the scanning rate of a host which has been infected for t time units. More precisely, we assume such a host scans the total address space on average at a rate of $v\theta(t)$ connection attempts per second. Under these assumptions, we can show that the infected fraction $r(t)$ evolves according to the differential equation:

$$\frac{dr}{dt} = \alpha v(1-r(t))\left[r(0)\theta(t) + \int_0^t r'(s)\theta(t-s)ds\right].$$

Indeed, the number of hosts infected in some small time interval $[s, s+ds]$ is $r'(s)ds$. At time $t > s$, these scan on average at a rate of $v\theta(t-s)$. Such scans are targeted at vulnerable addresses with probability α; the targets have not yet been infected with probability $1 - r(t)$. Summing over all such intervals $[s, s+ds]$, and accounting for infections due to initially infected hosts, yields this equation.

We are interested only in the behaviour of epidemics until $r(t)$ reaches some small δ, which we take to reflect global epidemic outbreak; think of δ being 5% for instance. We may thus consider the tight upper bound to $r(t)$, which solves the linear equation obtained by suppressing the factor $(1 - r(t))$ in the above. This now reads

$$r'(t) = v\alpha\left[r(0)\theta(t) + \int_0^t r'(s)\theta(t-s)ds\right].$$

Upon integrating, this reads

$$r(t) = r(0) + \int_0^t r(t-s)v\alpha\theta(s)ds. \qquad (1)$$

This is a classical equation, known as a renewal equation; see e.g. [1], Chapter V[1]. The epidemics will spread or die out exponentially fast depending on whether the integral $\int_0^\infty v\alpha\theta(s)ds$ is larger or smaller than one. This integral yields the mean number of hosts infected by a single initial infective, which is referred to as the basic reproduction number in epidemiology. In the critical case where this integral equals one, for large T, we have by the renewal theorem the estimate

$$r(T) \sim \frac{r(0)T}{v\alpha\int_0^\infty s\theta(s)ds}. \qquad (2)$$

In the subritical case (i.e. when $v\alpha\int_0^\infty \theta(s)ds < 1$), then for large T one has that

$$r(T) \sim \frac{r(0)}{1 - v\alpha\int_0^\infty \theta(s)ds}. \qquad (3)$$

In the supercritical case when $v\alpha\int_0^\infty \theta(s)ds > 1$, let $\nu > 0$ be such that

$$v\alpha\int_0^\infty e^{-\nu s}\theta(s)ds = 1. \qquad (4)$$

Then one has the estimate for large T:

$$r(T) \sim e^{\nu T}\frac{r(0)}{v\alpha\int_0^\infty se^{-\nu s}\theta(s)ds}. \qquad (5)$$

The parameter ν is called the exponent of the epidemics; it determines the *doubling time* of the epidemics, which reads $\nu^{-1}\ln(2)$.

These general results indicate that, in order for the epidemics to remain controlled in a time interval of length T the epidemics have to be either subcritical, or supercritical with a doubling time of the order of T. Indeed, in the supercritical case, $r(T)$ may still be small provided νT is small.

In the present context, a design objective for countermeasures is to prevent global spread before a vulnerability is identified, patches are developped and distributed. We may then take T to be of the order of days.

It should be noted that the above framework can be extended to consider more general scanning rate profiles, e.g. captured by a scanning speed $v(t)$ depending on the time t since infection. All of the above discussion carries over to this case if we introduce the function $S(t)$ of scanning attempts that can be made within t time units following infection, which summarizes the interaction between the worm and countermeasure, and if we replace in the above equations $v\theta(t)$ by the derivative $S'(t)$.

We now apply this general framework to specific scenarios, by identifying the function $\theta(t)$ corresponding to specific countermeasures.

B. Quarantining mechanisms

We first consider the case of quarantining mechanisms, in which a host's connection attempts are unaffected until some random time when it is completely blocked. Let τ denote the random time it takes after infection for quarantining to take place. Then the rate at which a host which has been infected for t seconds issues scans is

[1] This derivation is similar to that of Lotka's integral equation of population dynamics; see [1], p.143.

given by $v\mathbf{P}(\tau \geq t)$, where $\mathbf{P}(\tau \geq t)$ denotes the probability that the random detection time is larger than or equal to t. This corresponds to setting $\theta(t) = \mathbf{P}(\tau \geq t)$. As $\int_0^\infty \theta(t)dt$ equals the average detection time, denoted $\overline{\tau}$, the general results described above imply that the epidemic will be subcritical, critical or supercritical as $\alpha v \overline{\tau}$ is less than, equal to or greater than one respectively.

C. Throttling mechanisms

We now consider mechanisms where, which reduce the rate at which a host can make new connections when it is deemed suspicious. This is expected to effectively shut down the host if scanning is at a high rate. In the case of low scanning rate, this could provide benefits without incurring the penalty of shutting down the host.

a) Williamson's throttle: Consider Williamson's throttle, as proposed in [17]. According to this scheme, a host's connection requests go through a FIFO queue, and are processed at rate c connections per second. Assuming the host generates w non-wormy connection attempts per second, we find that the worm effectively scans at a rate of $vc/(v+w)$ if it competes fairly with the host's legitimate traffic, and if the submitted rate $v + w$ is larger than c. The slow-down factor is then $\theta(t) = c/(v + w)$, as long as the host is not blocked.

Assume now, as in Williamson's proposal, that the host is quarantined when the queue reaches some critical level q. The time it takes to reach that level is $\tau_{fill} = q/(v+w-c)$. We thus have $\theta(t) = c/(v+w)$ if $t \leq \tau_{fill}$, and $\theta(t) = 0$ otherwise. The quantity characterising criticality of the epidemics then reads

$$\alpha v \int_0^\infty \theta(t)dt = \alpha v \frac{c}{v+w} \tau_{fill}.$$

In the supercritical case, Equation (4) again characterises the epidemics' exponent.

b) Max-Failure throttle: The Max-Failure (MF) scheme that we now propose is meant to throttle connection attempts to ensure that the long-run rate of failed connection attempts is below some pre-specified target rate, λ_f. It is similar in spirit to the Leaky Bucket traffic scheduler (see e.g. [8], [3] and references therein), and enjoys similar optimality properties. It works as follows. Connection attempts are fed to a queue, and need to consume a token to be processed. There is a token pool, with maximal size B; the pool is continuously replenished at a rate λ_f. Finally, when a previously made connection attempt is acknowledged, the token it has used is released and put back in the pool[2]. The appeal of MF is captured by the following proposition.

Proposition 1: Provided that successful connection attempts always get acknowledged less than $1/\lambda_f$ seconds after the connection request has been sent, under MF, in any interval $[s, t]$, the number of connection attempts that resulted in failure is at most $B + \lambda_f(t - s)$.

Furthermore, MF schedules connection requests at the earliest, among all schedulers that satisfy this constraint on the number of failed connection attempts in each interval $[s, t]$, and that do not have prior information on whether connection attempts will fail or not.

Proof: The first part is easily established, and we omit the detailed argument due to lack of space. For the second part, consider a process of connection requests, and an alternative scheme that would at some time t schedule a request that is backlogged at t under MF. Thus the MF token pool contains less than one unit at t. Equivalently, there exists some $s < t$ such that under MF, the number n of attempted and not yet acknowledged connection attempts made in interval $[s, t]$ verifies $n + 1 > B + (t - s)\lambda_f$. If these connection attempts, as well as the additional one made by the alternative throttle, all fail, then this throttle will experience at least $n + 1 > B + (t - s)\lambda_f$ failures within interval $[s, t]$, which establishes the claim. ■

Consider now hosts who submit non-wormy connection requests at a rate w, a proportion p_f of which fail. Assume as before that a worm submits scan attempts at a rate of v, a proportion $(1 - \alpha)$ of which are directed at non-vulnerable hosts. We make the additional assumption that scanning attempts result in failures when targeted at non-vulnerable hosts, so that for an infected host, the rate of failed connection attempts due to worm scanning is $v(1 - \alpha)$.

[2] Schechter et al. [13] have proposed a similar scheme, which differs in that they replace two instead of one token into the pool upon acknowledgement.

Assuming that the failure rate in the absence of throttling, namely $p_f w + (1-\alpha)v$, is greater than λ_f, then after infection the token pool is emptied in time $\tau_{empty} = B/(p_f w + (1-\alpha)v)$, after which connection attempts are slowed down by a factor of $\lambda_f/(p_f w + (1-\alpha)v)$. Thus, for the MF throttle, we have the characterization

$$\theta(t) = \begin{cases} 1 & \text{if } t \leq \tau_{empty}, \\ \frac{\lambda_f}{p_f w + (1-\alpha)v} & \text{otherwise.} \end{cases} \qquad (6)$$

Assuming further that, as in Williamson's throttle, the host gets blocked once there are q backlogged requests, we find that it takes

$$\tau'_{fill} = \frac{q}{(v+w)(1-\lambda_f/(p_f w + (1-\alpha)v))]}$$

seconds to quarantine the host once its token pool is exhausted. The quantity characterising criticality of the epidemics then reads

$$\frac{\alpha v B}{p_f w + (1-\alpha)v} + \frac{\lambda_f q (v+w)^{-1}}{p_f w + (1-\alpha)v - \lambda_f}.$$

The growth exponent can be calculated as before from Equation (4) in the super-critical case.

III. Optimal detection procedures

The aim of this section is to describe optimal tests for detecting infections under specific assumptions on the probabilistic model of a host's normal behaviour and worm behaviour. We consider detection procedures based on the observation of failed connection attempts only, beacuse it is straightforward for a worm to artificially increase the number of successful connections, and thus defeat detectors based on the ratio of failures to successes.

The model we take of normal behaviour is the following. A host attempts connections that fail at a rate λ_0, with the failure times being the points of a Poisson process. This assumption is made for the sake of tractability. It is however plausible if failures are rare, since Poisson processes are good approximations for processes of rare, independent instances of events.

If the host is infected, the worm scans occur at the points of an independent Poisson process of rate v. Again, this assumption is made for the sake of tractability. The worm scans are successful with probability α and the successes form an iid Bernoulli sequence. This is consistent with the assumption of a randomly scanning worm that picks target addresses uniformly at random from the whole address space. Hence, the aggregate failure process is Poisson with rate $\lambda_1 = \lambda_0 + (1-\alpha)v$.

We first assume that λ_0 and λ_1 are known, in order to get bounds on the best case achievable performance. We shall also discuss schemes that are close to optimal and do not rely on knowledge of these parameters in the next subsection.

A. Optimal CUSUM test for known scan rates

The objective is to detect that a host is infected in the shortest possible time after infection, while ensuring that the false alarm rate is acceptably low. This is a changepoint detection problem and it is known that the CUSUM strategy described below has certain desirable optimality properties (see, e.g., [9], [12]).

Note that this strategy is very similar to the detection mechanism proposed in [13]. Given a sequence of observed inter-failure times $t_1, t_2, t_3, \ldots$, we declare that the host is infected after the R-th failure time, where

$$R = \inf\Big\{ n : \max_{1 \leq k \leq n} \Big[\sum_{i=k}^{n} \log \frac{f_1(t_i)}{f_0(t_i)} \Big] \geq c \Big\}. \qquad (7)$$

Here, c is a specified threshold, and f_0 and f_1 are densities under the null hypothesis that the machine isn't infected and the alternative hypothesis that it is, i.e., $f_i(t) = \lambda_i e^{-\lambda_i t}$ for $t \geq 0$ and $i = 0, 1$.

In words, we declare the machine to be infected when the log likelihood ratio of its being infected to its being uninfected over some finite interval in the past exceeds a threshold, c. In order to implement the test, we do not need to maintain the entire history of observations and compute the maximum over $k \leq n$ as above. The same quantity can be computed recursively as follows. Define $Q_0 = 0$, and for $i \geq 1$,

$$Q_i = \max\Big\{ 0, Q_{i-1} + \log \frac{f_1(t_i)}{f_0(t_i)} \Big\}. \qquad (8)$$

The above recursion is known as Lindley's equation and its solution is

$$Q_n = \max\left(0, \max_{1\le k\le n}\Big[\sum_{i=k}^{n} \log\frac{f_1(t_i)}{f_0(t_i)}\Big]\right)$$

$$= \max_{0\le k\le n}\left[(n-k)\log\frac{\lambda_1}{\lambda_0} - (\lambda_1 - \lambda_0)(T_n - T_k)\right] \quad (9)$$

where $T_n = \sum_{i=1}^{n} t_i$ is the time of the n^{th} failure. Now, by (7), $R = \inf\{n : Q_n \ge c\}$.

In the literature on change point detection problems, the random variable R is called the run length. The results in [9] imply that the CUSUM test is optimal in that it minimizes the expected time between infection and detection, given a target *false positive rate*, characterised by the average run length $E[R]$ when the host is uninfected. We now obtain an estimate of this quantity.

Observe that the process Q_i is a random walk reflected at 0; under the null hypothesis that the machine is uninfected, the free random walk (without reflection) has negative drift. We want to compute the level-crossing time for this random walk to exceed a level c. Using standard large deviation estimates for reflected random walks, it can be shown that, for large c,

$$\frac{1}{c}\log E_0[R] \sim \theta^* := \sup\{\theta > 0 : M(\theta) \le 1\}, \quad (10)$$

where

$$M(\theta) = E_0\Big[\exp\Big(\theta \log\frac{f_1(t_i)}{f_0(t_i)}\Big)\Big].$$

Here, we use the subscript 0 to denote that expectations are taken with respect to the distribution of t_i under the null hypothesis, i.e., t_i are i.i.d., exponentially distributed with mean $1/\lambda_0$. A straightforward calculation yields

$$M(\theta) = \begin{cases} \frac{\lambda_0}{\lambda_0 + \theta(\lambda_1 - \lambda_0)}\left(\frac{\lambda_1}{\lambda_0}\right)^{\theta} & \text{if } \theta > -\frac{\lambda_0}{\lambda_1 - \lambda_0} \\ +\infty & \text{otherwise.} \end{cases}$$

From this, it can be readily verified that $\theta^* = 1$.

We may wish to choose the threshold c so that the false alarm probability within some specified time interval T, say one month or one year, is smaller than a specified quantity ϵ. This is equivalent to requiring that $P(R \le \lambda_0 T) < \epsilon$. Using the so-called Kingman bound (see e.g. [2])

$$P(Q_i \ge c) \le e^{-\theta^* c} \text{ with } \theta^* = 1,$$

and the union bound

$$P(R \le \lambda_0 T) \le \sum_{i=1}^{\lambda_0 T} P(Q_i \ge c) \le \lambda_0 T e^{-\theta^* c},$$

we find that the constraint $P(R \le \lambda_0 T) < \epsilon$ is satisfied provided

$$c \ge \log\frac{\lambda_0 T}{\epsilon}. \quad (11)$$

B. Subptimal detectors for unknown scan rates

In this section, we ask the following question: given a bound ν on the acceptable growth rate of the epidemic, is it possible to design a detector that restricts the growth rate to no more than ν, while simultaneously ensuring that the false alarm probability over a specified time window T doesn't exceed a specified threshold ϵ? The answer would define a feasible region that imposes fundamental limits on detector performance.

In the most general setting, we would allow for all possible worm scanning profiles, but we restrict the strategy space by considering only constant scanning rates. Thus, the worm can choose an arbitrary scanning rate z; then, for each t, it makes zt scanning attempts in the first t time units after infecting a host. Of these, $(1-\alpha)zt$ are failures on average.

To start with, we consider general detector profiles. We shall then show that the optimal detector is well approximated by a CUSUM detector. Now, an arbitrary detector can be specified by a function $F(\cdot)$ which determines the maximum number of failed connection attempts possible in any contiguous time window before the host is quarantined. Given $F(\cdot)$ and z, the maximum scanning rate possible at time t is given by

$$\theta(t) = \begin{cases} z, & t \le t_z, \\ 0, & t > t_z, \end{cases}$$

where $t_z \ge 0$ solves $F(t_z) = (1-\alpha)zt_z$; take $t_z = \infty$ if this equation has no solution. Substituting this in (4), we find that the growth exponent is bounded above by ν provided the following inequality is satisfied:

$$\alpha \int_0^{t_z} z e^{-\nu t} dt \le 1, \text{ i.e., } 1 - e^{-\nu t_z} \le \frac{\nu}{\alpha z}.$$

We want this inequality to hold for every z. Now, each choice of z yields a t_z as noted above. Hence, expressing z in terms of $F(\cdot)$ and t_z, we can rewrite the above inequality as

$$1 - e^{-\nu t} \leq \frac{(1-\alpha)\nu t}{\alpha F(t)} \qquad (12)$$

or equivalently,

$$F(t) \leq \frac{1-\alpha}{\alpha} \frac{\nu t}{1 - e^{-\nu t}}, \qquad (13)$$

for all positive t. The above inequality yields a necessary and sufficient condition for a detector to guarantee the bound ν on the worm growth exponent, for arbitrary choice of worm scanning rate z. In other words, defining $F(\cdot)$ with equality above yields the *optimal detector* in the following sense: any detector which guarantees a worm growth exponent no bigger than ν against constant rate worms with arbitrary scanning rate will have at least as large a false alarm rate.

The optimal detector has the property that the worm growth exponent is insensitive to the scanning rate (so long as the scanning rate is large enough that it will eventually result in quarantine). Thus, no cleverness is required on the part of the worm designer if the optimal detector is employed. Note that we are implicitly considering a game between detectors and worms, where the worm has second mover advantage. This is a realistic assumption in a scenario where there is a large installed base of worm detection software, and worm designers could use their knowledge of this software in specifying worm parameters.

Now, we could compute the false alarm rate for the optimal detector as defined above. However, in practice, it is not straightforward to enforce an envelope with arbitrary shapes as above, whereas we saw that an affine envelope corresponds simply to a queue. Therefore, we first find an affine function that is a lower bound to the RHS of (13). It is readily verified that

$$F(t) = \frac{1-\alpha}{\alpha}\left[1 + \frac{\nu t}{2}\right] \qquad (14)$$

is such a lower bound. In particular, using $F(\cdot)$ specified by (14) guarantees that the worm growth rate is smaller than ν, since $F(\cdot)$ imposes a more stringent bound on the number of failed connections than the optimal detector.

Combining the specific choice for F given in (14) together with the condition for achieving target false alarm probabilities of ϵ every T time units given in (11), we arrive at the following:

Proposition 2: Let λ_0 be the rate at which hosts fail due to non-wormy traffic, and let α be a known bound on the fraction of address space occupied by vulnerable hosts. Let ϵ be the target probability of false alarms over a time horizon of T. Then one can tune the parameters of CUSUM detectors so that the exponent of the epidemics is not larger than

$$\nu_{\max} = 2\lambda_0 \frac{1-2\alpha}{(1-\alpha)c}\left[\exp\left(\frac{\alpha c}{1-2\alpha}\right) - 1\right] \qquad (15)$$

for any worm that scans at constant rate. The CUSUM detector satisfying these properties is characterized by the choice of c as in Equation (11), and the following choice of λ_1:

$$\lambda_1 = \lambda_0 \exp\left(\frac{\alpha c}{1-2\alpha}\right) \cdot \qquad (16)$$

Conversely, for any detector guaranteeing that no constant scanning rate can achieve a growth exponent bigger than $\nu_{max}/2$, the false alarm probability over time horizon T is at least ϵ.

Remark 1: Note the slackness between the first statement and its converse: by restricting to CUSUM detectors, we are losing at most a factor of two in the maximal achievable growth exponent.

Proof: Observe from (7) and (9) that the maximum number of failed connection attempts allowed by a CUSUM detector up to time t, denoted $n(t)$, must satisfy

$$[n(t) - 1]\log\frac{\lambda_1}{\lambda_0} - (\lambda_1 - \lambda_0)t \leq c. \qquad (17)$$

This also reads

$$n(t) \leq 1 + \frac{c}{\log(\lambda_1/\lambda_0)} + \frac{\lambda_1 - \lambda_0}{\log(\lambda_1/\lambda_0)}t.$$

Thus, if we identify the coefficients in this affine constraint with those of the function F in (14), by the preceding discussion it is guaranteed that no constant scanning worm with success rate below α can achieve an exponent bigger than ν. This identification of coefficients yields

$$\begin{cases} \frac{1-\alpha}{\alpha} & = & 1 + \frac{c}{\log(\lambda_1/\lambda_0)}, \\ \frac{1-\alpha}{\alpha}\frac{\nu}{2} & = & \frac{\lambda_1 - \lambda_0}{\log(\lambda_1/\lambda_0)}. \end{cases}$$

The first relation is equivalent to (16), while the second one gives the value of ν as in (15).

Conversely, note that the affine function $F(t)$ as in (14) is larger than the function

$$G(t) := \frac{1-\alpha}{\alpha} \frac{(\nu/2)t}{1-e^{-(\nu/2)t}} \ ,$$

that is the right-hand side of Equation (13) with ν replaced by $\nu/2$. Hence, the affine constraint is weaker than the optimal detector for ensuring the bound $\nu/2$ on the growth rate, and thus incurs a lower false positive rate. ■

IV. Experiments

We now describe experiments exploiting a traffic trace, which was collected over 9 days at a corporate research facility comprising of the order of 500 workstations and servers. The IP and TCP headers were captured for all traffic crossing VLANs on the on-site network (i.e. between machines on-site). The traffic capture (using the NetMon2 tool) was post-processed to extract TCP connection attempts. We consider only TCP connections internal to the corporate environment in what follows. Hence the experiments to be described reflect the ability of worm detectors to contain the spread of worms inside a corporate environment.

Figure 1 represents the cumulative distribution functions of the number of destinations to which sources experience failed connection attempts. There is one curve per day of the trace. We observe that 95% of sources fail to less than 20 destinations each day, and all sources fail to less than 45 distinct destinations each day. The median number of distinct addresses to which sources fail is between 2 and 5 per day.

In a practical implementation of the CUSUM detector, we would maintain a *white list* of recently used addresses, as in the previous proposals [17], [13]. Connection attempts to addresses in the white list bypass the detector. Thus, repeated failures to the same address would be counted as one failure at the detector. Under such a scheme, failures seen by the detector are expected to be rare, and uncorrelated. This provides justification for the assumption that failures follow a Poisson process.

Figure 2 shows the cumulative distribution function per source of the total number of failed connection attempts that would be measured by a CUSUM detector over the whole 9-day trace duration.

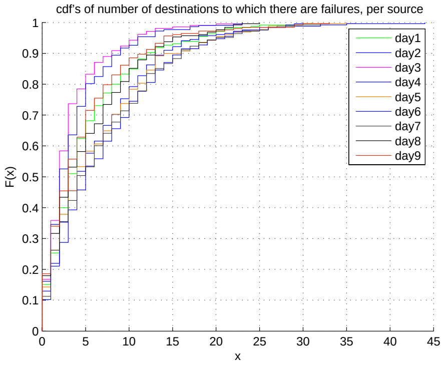

Fig. 1. Cumulative distribution function of numbers of destinations to which sources fail, from corporate trace.

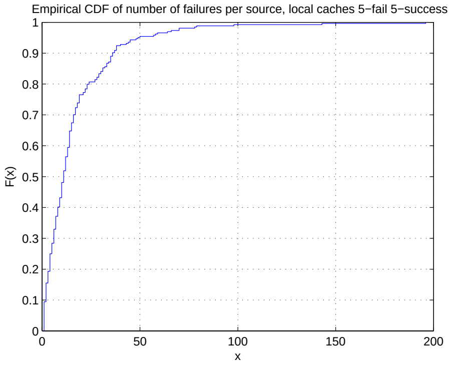

Fig. 2. Cumulative distribution function of numbers of failures per source, using white lists, over 9 days.

This figure is produced with a combined white list mechanism, comprising the last five addresses to which there have been failed connection attempts, and the last five addresses to which there have been successful connection attempts. These are easily maintained by using the LRU (Least Recently Used) replacement strategy. W e observe that over 9 days, 95% of sources would have less than 50 failures to addresses not in the cache, while no source ever experiences more than 200 failures. This suggests that we chose background failure rate λ_0 of the order of one failure per hour[3].

We now assess how CUSUM detectors would perform in such a corporate environment. Specifically, we fix a target false positive probability $\epsilon = e^{-1}$ per year[4]. We then use the CUSUM detector described in Proposition 2, which we know will guarantee a bound on the growth exponent not larger than twice the best possible bound, given the false positive constraint. We then evaluate the largest possible growth rate for this CUSUM detector from Equation (15). Figure 3 plots the corresponding time for the worm to grow ten-fold, that is $\log(10)/\nu_{max}$ versus the bound α on the fraction of vulnerable hosts.

We see that ten-fold increase will take of the order of more than a day provided α is less than 0.04. Thus, in networks with at most this fraction of vulnerable hosts, this kind of detectors will slow down epidemics sufficiently for human intervention, e.g. development and distribution of patches, to take over.

Recently, automatic patch generation and distribution schemes have been proposed (see [4]) which may have reaction times of the order of tens of seconds. As we see from Figure 3, when α is as large as 0.25, our detectors can force the time to a ten-fold increase to be larger than six minutes, which is well within the scope of such schemes.

To complement these analytical evaluations, we have run trace-driven simulations, with one initially infected host scanning randomly, and successfully finding a victim 25% of the time, and where each host runs the CUSUM detector tuned as in Proposition 2, with a failure rate λ_0 of once per hour, and a target false positive rate as above.

Figure 4 illustrates the cumulative distribution function of the number of hosts eventually infected after 30 minutes, for various worm scanning speeds. Each curve is obtained from 40 simulation runs. We observe that with the slower scanning speed of 0.01 attempts a second, the spread is consistently larger than with the higher scanning speeds of 0.1 and 1 attempts per second. With a scanning speed of 10, in all simulation runs the initially infected hosts got blocked before contaminating anyone else. The number of infected hosts after 30 minutes is, in all these cases, highly random: otherwise, the cumulative distribution functions would degenerate to step functions.

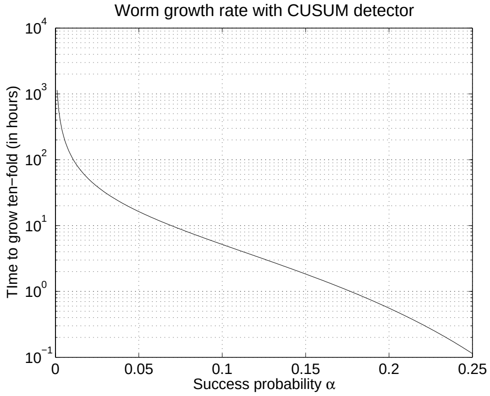

Fig. 3. Time (in hours) for a ten-fold growth in the number of infected hosts as a function of the fraction α of vulnerable hosts.

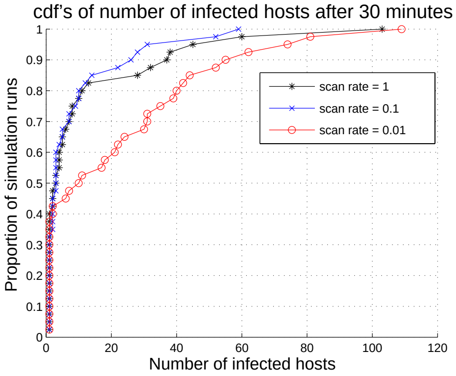

Fig. 4. Cumulative distribution function of number of infected hosts after 30 minutes, starting from one infected host, for $\alpha = 0.25$.

Table 1 provides the fraction of simulation runs where the worm spread was stopped by 30 minutes, and the average number of infected hosts among the runs where there still remained infected hosts (growth factor). These again suggest that the epidemics was supercritical for a scanning rate of 0.01, with a growth factor of around 34 per 30 minutes, while for scanning rates of 0.1 and 1,

[3] Clearly, λ_0 may be different in other, non-corporate environments; for example, there were no P2P applications in the trace we studied.

[4] As the time to a false alarm is roughly exponentially distributed, this choice is consistent with an average of one false alarm per year.

scanning speed	0.01	0.1	1
epidemics stopped	40%	90%	85%
growth factor	33.8	1	2

Table 1: Fraction of blocked epidemics, and average growth factor after 30 minutes.

the epidemics was subcritical.

In the next Section we propose techniques for coordinating the detectors at the individual hosts. The approach proposed in Section V-A has been assessed using trace-driven simulations. Figure 5 shows the cumulative distribution functions of the number of infected hosts after 30 minutes, for the same density $\alpha = 0.25$ of vulnerable hosts as before, and evaluated over 40 simulation runs for each curve. As in the absence of coordination, we find that the slower the scanning speed, the more successful the infection. However, notice the reduction in the total number of infected nodes. Also, in all simulation runs, no infected hosts remained unquarantined after 30 minutes.

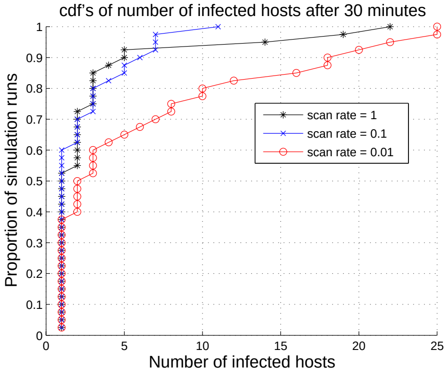

Fig. 5. Cumulative distribution function of number of infected hosts after 30 minutes, starting from one infected host, for $\alpha = 0.25$, with coordination. Parameters for Beta prior: (500,2).

V. A FRAMEWORK FOR COORDINATION

In this section, we first describe our proposal for coordinating individual CUSUM detectors. We then present two possible implementations. Finally we sketch an approach for learning individual hosts' failure rates and explain how the techniques considered so far extend to heterogeneous environments.

A. A Bayesian version of CUSUM

The CUSUM detector applied the following test: quarantine the host if

$$Q_n := \sum_{i=1}^{n} \log \frac{f_1(t_i)}{f_0(t_i)} > c \qquad (18)$$

for any n, where t_i are the past inter-failure times, f_1 and f_0 are the densities of inter-failure times for infected and healthy hosts. The threshold c determines the trade-off between false alarm probability and the probability of failing to detect an infection. This trade-off can be cast in a Bayesian decision theoretic framework. Let p denote our prior belief that the host was infected during the past n failure times. The posterior probability that it was infected is given by

$$\begin{aligned}\hat{p} &= \frac{p\prod_{i=1}^{n} f_1(t_i)}{p\prod_{i=1}^{n} f_1(t_i)} + (1-p)\prod_{i=1}^{n} f_0(t_i) \\ &= \frac{pe^{Q_n}}{1-p+pe^{Q_n}}.\end{aligned} \qquad (19)$$

We associate a cost C_Q with the decision to quarantine a host if it is actually healthy, and a cost C_0 with the decision not to quarantine it if it is actually infected. Correct decisions incur zero cost. The objective is to minimise the expected cost. Clearly, this is achieved by quarantining if and only if $\hat{p}C_0 > (1-\hat{p})C_Q$. Substituting for $\hat{p}$ from above and simplifying, we can state the condition for quarantining as $\frac{p}{1-p}e^{Q_n} > C_Q/C_0$, or equivalently,

$$Q_n > \log \frac{(1-p)C_Q}{pC_0}. \qquad (20)$$

Comparing this with (18), we see that the Bayesian decision theoretic approach yields the same test provided we identify c with $(1-p)C_Q/(pC_0)$.

Thus, the approach outlined in Section III corresponds to each host implementing a Bayesian decision procedure in isolation. Can they do better by cooperating and pooling their observations? A natural way to do this is to make p a common parameter for all hosts, denoting the fraction of hosts which are infected. Since this is unknown, we describe our uncertainty through a prior distribution $f(\cdot)$ for the parameter p.[5] For the sake

[5] We could have also used a prior for p in the individual case discussed above, but the conclusion there would have been identical with p being the prior mean.

of tractability, we take f to be a Beta distribution with parameters α, β; that is to say, for $p \in [0,1]$,

$$f(p) = f_{\alpha,\beta}(p) = \frac{1}{\mathcal{B}(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \quad (21)$$

where $\mathcal{B}(\alpha,\beta)$ is a normalising constant.

Let X_i denote the indicator that host i is infected, i.e., $X_i = 1$ if host i is infected and 0 if it is healthy. Now, conditional on p, each host is assumed to have equal probability p of being infected, independent of all other hosts. Next, conditional on X_i, the inter-failure times at host i are assumed to be i.i.d. with distribution $f_{X_i}(\cdot)$, and independent of inter-failure times at other hosts. Expressing this formally, the model specifies the following joint distribution for infection probability, infection status and inter-failure times:

$$f(p,\mathbf{X},\mathbf{t}) = f(p)\prod_{i=1}^{N} p^{X_i}(1-p)^{1-X_i}\prod_{j=1}^{n} f_{X_i}(t_j^i), \quad (22)$$

where t_j^i denotes the j^{th} inter-failure time at host i. From this, we can compute the conditional distributions of p and X_i given the observed inter-failure times at different hosts.

In the uncoordinated case studied earlier, inference about X_i was based only on observations of failure times at host i; now, failure times at other hosts could influence the posterior distribution of X_i via their effect on the posterior distribution of p. Is this desirable? We claim that the answer is yes; when we have evidence that there is a worm outbreak in the population, then we should be more suspicious of unusual failure behaviour at any individual host. The Bayesian formulation described above fleshes out this intuition and gives it a precise mathematical description.

Define

$$Q_n^i = \sum_{j=1}^{n} \log \frac{f_1(t_j^i)}{f_0(t_j^i)} \quad (23)$$

to be the log-likehood ratio of the observed inter-failure times at host i between the hypotheses that it is and isn't infected. We then have:

Proposition 3: The posterior distribution of the infection probability p is given by

$$f(p|\mathbf{t}) = \frac{1}{Z} f(p)\prod_{i=1}^{n}\Big(1-p+pe^{Q_n^i}\Big), \quad (24)$$

where Z is a normalising constant that does not depend on p. If f is the Beta distribution (21) and the parameters α, β are bigger than 1, then the posterior mode is at the unique solution in $(0,1)$ of the equation

$$\frac{\alpha-1}{p} - \frac{\beta-1}{1-p} + \sum_{i=1}^{N} \frac{e^{Q_n^i}-1}{(1-p)+pe^{Q_n^i}} = 0. \quad (25)$$

The proof is omitted for lack of space.

Now, conditional on p and the observed failure times at host j, the posterior probability that $X_j = 1$, i.e., that host j is infected, is given, as in (19), by the expression

$$P(X_j = 1|p,\mathbf{t}) = \frac{pe^{Q_n^j}}{1-p+pe^{Q_n^j}}.$$

We can remove the conditioning on p by integrating the above expression with respect to the posterior distribution. However, this is not analytically tractable. Therefore, we adopt the expedient of simply replacing p with its posterior mode, denoted $\hat{p}$ and given by the solution of (25). This can be heuristically justified on the grounds that as the number of nodes and hence the total number of observations grows large, the posterior becomes increasingly concentrated around its mode. Thus, we rewrite the above as

$$P(X_j = 1|\mathbf{t}) \approx \frac{\hat{p}e^{Q_n^j}}{1-\hat{p}+\hat{p}e^{Q_n^j}},$$

where $\hat{p}$ solves (25). Letting C_Q and C_0 denote the costs of incorrectly quarantining and failing to quarantine this host respectively, we obtain as before the Bayes decision procedure: quarantine host j if and only if

$$Q_n^j > \frac{(1-\hat{p})C_Q}{\hat{p}C_0}. \quad (26)$$

Comparing this with (20), we note that the only difference is that p has been replaced by $\hat{p}$. Therefore, the decision rule can be implemented by using a CUSUM detector at each node j, but with CUSUM threshold c_{old} replaced by

$$c_{new} = c_{old} + \log\frac{1-\hat{p}}{\hat{p}} - \log\frac{1-p}{p}; \quad (27)$$

this is clear from (23). Here p denotes the mode of the prior, which is equal to $(\alpha-1)/(\alpha+\beta-2)$ for the assumed Beta prior.

B. Implementation

The above scheme requires each host to update its CUSUM threshold based on the revised estimate $\hat{p}$ of the infection probability, that is the mode of the posterior distribution of p, and thus the solution of (25). There are a number of ways that the coordination can be done in practice: we can have a central node (or chosen host) receive the CUSUM statistics Q^i from the hosts periodically, calculate $\hat{p}$ from (25) and send this back to the hosts. The numerical calculation of $\hat{p}$ is not expensive, and can be done efficiently by bisection search, for example.

Alternatively, the hosts are divided into domains d_j, with a leader j responsible for that domain. Host j periodically collects CUSUM statistics from its domain, exchanges messages with the other dedicated hosts for producing the revised estimate $\hat{p}$, and feeds it back to hosts in its domain. We now describe a strategy for distributed computation of $\hat{p}$ among leaders, which does not require them to broadcast the collection of CUSUM statistics they are responsible for. Instead, they broadcast two summary statistics periodically. The summary statistics for domain j are its size $|d_j|$ (in number of hosts) and the following quantity:

$$s_{n,j} := \sum_{i \in d_j} \frac{p_n e^{Q^i}}{1 - p_n + p_n e^{Q^i}}.$$

There, p_n is the current estimate of $\hat{p}$ after n communication rounds between leaders. It is updated at each leader after the $s_{n,j}$'s and $|d_j|$'s are broadcast according to

$$p_{n+1} = \frac{\alpha - 1 + \sum_j s_{n,j}}{\sum_j |d_j| + \alpha + \beta - 2}.$$

It can be shown that under the assumptions $\alpha, \beta > 1$, the iterates p_n converge to a fixed point $\hat{p}$ which satisfies (25). As convergence is exponentially fast, only few iterations are needed. Hence we obtain a significant gain on communication cost as compared to the centralised approach. The proof of convergence of the iterative scheme is omitted due to lack of space.

C. Heterogeneous hosts, unknown parameters and learning

In the discussion above, the failure time distributions f_1 and f_0 for infected and healthy hosts were assumed to be the same at all hosts, and to be known. These assumptions were made for clarity of exposition, but are not necessary. We now briefly sketch how these assumptions can be relaxed.

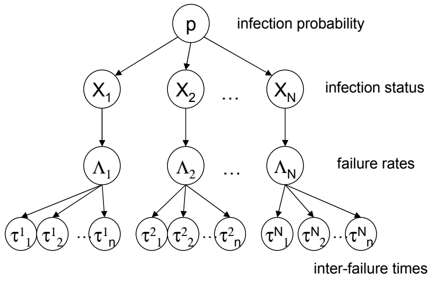

Fig. 6. Belief network model for inter-failure times.

In general, each host i can have a different distribution for inter-failure times, f_0^i, and a different distribution for failures caused by a worm f_1^i. The framework we use is a Bayesian belief network where the relationship between the variables is shown in Figure 6. The top p node in the belief network of Figure 6 plays the role of correlating the observations across hosts. If the inter-failure times distributions f_1, f_0 are taken to be exponential distributions with parameters λ_1, λ_0 respectively, then the failure rates Λ_i are just λ_{X_i}.

In general there is some uncertainty over the natural failure rate Λ_i of the host, and a worm's scanning rate. This can be incorporated into the belief network by assuming that the failure rate Λ_i itself is unknown. We assume that, conditional on the value of X_i, it is distributed according to a Gamma distribution with scale parameter θ and shape parameter η which depend on the value of X_i. That is to say:

$$\frac{d}{dx}\mathbf{P}(\Lambda_i \leq x | X_i) = \gamma_{\theta(X_i),\eta(X_i)}(x),$$

where $\gamma_{\theta,\eta}(x) = \mathbf{1}_{x \geq 0} e^{-\theta x} \theta^\eta x^{\eta-1} \frac{1}{\Gamma(\eta)}$. This choice of a Gamma prior distribution is made to ensure tractability of posterior distribution evaluation. Conditional on the parameter Λ_i, the inter-failure times τ_j^i are independent and identically exponentially distributed with parameter Λ_i.

The analysis of Section V-A can be pushed through. Essentially all the formulas have the same form, though we have to take expectations over the unknown parameters. Instead of defining Q_n^i as in (23) we have

$$Q_n^i = \log \frac{E_{\gamma_{\theta(1),\eta(1)}}\left[\prod_{j=1}^n \Lambda_i e^{-\Lambda_i t_j^i}\right]}{E_{\gamma_{\theta 0,\eta(0)}}\left[\prod_{j=1}^n \Lambda_i e^{-\Lambda_i t_j^i}\right]}$$

This corresponds to a generalisation of the CUSUM statistic. In fact with our choice of prior for Λ_i this has the explicit form

$$Q_n^i = \log \frac{\rho_i(1)}{\rho_i(0)} \qquad (28)$$

where

$$\rho_i(X) := \frac{\theta(X)^{\eta(X)}\Gamma(\eta(X)+n)}{(\theta(X)+S_i)^{n+\eta(X)}\Gamma(\eta(X))}. \qquad (29)$$

and $S_i := \sum_{j=1}^n \tau_j^i$.

VI. Concluding remarks

In this paper we have used epidemiological modeling to characterize the interplay between scanning worms and countermeasures. We have illustrated this on both existing throttling schemes and the novel Max-Failure scheme, whose optimality properties have been identified (Proposition 1). We have then identified optimal quarantining profiles, for which worm spread is independent of worm scanning speed. This has allowed us to determine simple CUSUM tests that ensure that worm spread is at most twice as fast as what an optimal quarantining profile can ensure (Proposition 2). Our approach to the design of countermeasures, focusing on the exponent of potential epidemics, allows the derivation of explicit designs, and thus reduces the number of free parameters to be configured.

We have proposed a Bayesian approach for combining individual detectors and hence speed up reaction of such detectors to worms. The uncoordinated and coordinated approaches have been evaluated using trace-driven simulations. The experimental results indicate that the proposed countermeasures can be efficient in environments such as the corporate research lab where the trace has been collected. We intend to experiment our proposals in different environments, targetting more particularly home users. We also plan to evaluate more thoroughly the Bayesian approach for learning individual host characteristics sketched in Section V-C.

References

[1] S. Asmussen, *Applied Probability and Queues*. Springer-Verlag, 2003.

[2] F. Baccelli and P. Brémaud, *Elements of Queueing Theory*. Springer-Verlag, 2003.

[3] C.-S. Chang, *Performance Guarantees in Communication Networks*, Springer, 2000.

[4] M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham. Vigilante: End-to-End Containment of Internet Worms. In *Proc. SOSP 2005*, Brighton, United Kingdom, October 2005.

[5] D.J. Daley and J. Gani, *Epidemic Modelling: An Introduction*, Cambridge University Press, 1999.

[6] J. Jung and V. Paxson and A. Berger and H. Balakrishnan, Fast portscan detection using sequential hypothesis testing, In Proc. IEEE Symposium on Security and Privacy, May 9– 12, 2004.

[7] H.A. Kim and B. Karp, Autograph: toward automated, distributed worm signature detection. In Proc. 13th USENIX Security Symposium, August 2004.

[8] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, Springer, 2001.

[9] G. Moustakides, Optimal procedures for detecting changes in distributions. Annals of Statistics, vol. 14, pp. 137-1387, 1986.

[10] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, Inside the Slammer Worm. IEEE Security and Privacy, 1(4), pp. 33-39, 2003.

[11] D. Moore, C. Shannon and J. Brown, Code-Red: a case study on the spread and victims of an Internet Worm. In Proc. ACM/USENIX Internet Measurement Workshop, 2002.

[12] Y. Ritov, Decision theoretic optimality of the CUSUM procedure, Annals of Statistics, vol. 18, pp. 1464-1469, 1990.

[13] S.E. Schechter, J. Jung and A.W. Berger, Fast Detection of Scanning Worm Infections, 7th International Symposium on Recent Advances in Intrusion Detection (RAID), 2004.

[14] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis*, Chapman&Hall, London, 1995.

[15] J. Twycross and Matthew Williamson, Implementing and testing a virus throttle, In Proc. 12th USENIX Security Symposium, USENIX, 2003.

[16] N. Weaver, V. Paxon, S. Staniford and R. Cunningham, A taxonomy of computer worms. In Proc. ACM workshop on Rapid Malcode (WORM), 2003.

[17] M. M. Williamson, Throttling Viruses: Restricting Propagation to Defeat Mobile Malicious Code. In ACSAC, 2002.

[18] C. Zou, L. Gao, W. Gong and D. Towsley, Monitoring and Early Warning for Internet Worms, In Proc. of the 10th ACM Conference on Computer and Communications Security, pp. 190-199, 2003.